TECHNIQUES AND ISSUES IN MULTICAST SECURITY

Peter S. Kruus
Naval Research Laboratory
Washington, DC 20375
kruus@itd.nrl.navy.mil

Joseph P. Macker
Naval Research Laboratory
Washington, DC 20375
macker@itd.nrl.navy.mil

ABSTRACT

Multicast networking support is becoming an increasingly important future technology area for both commercial and military distributed and group-based applications. Integrating a multicast security solution involves numerous engineering tradeoffs. The end goal of effective operational performance and scalability over a heterogeneous internetwork is of primary interest for widescale adoption and application of such a capability. Various techniques that have been proposed to support multicast security are discussed and their relative merits are explored.

INTRODUCTION

Multicast communication as defined in [1] is an efficient means of distributing data to a group of participants. In contrast to unicast communications, multicast routing permits a single IP datagram to be routed to multiple hosts with minimal redundant transmission within a network. Membership in a multicast group is often highly dynamic, with receivers entering and leaving the multicast session without the permission or explicit knowledge of other hosts. The inherent cost and resource benefits of multicast routing and data delivery are clear; however, the group-oriented communication paradigm presents new and unique technical challenges beyond traditional network security approaches.

Potential security threats to multicast communications are similar to those encountered in unicast transmissions. Threats include the unauthorized creation, alteration, destruction, and illegitimate use of data [5]. In the case of multicast traffic, because of the inherent broad scope of a multicast session, the potential for attacks may be greater than for unicast traffic. It is desirable to secure these vulnerabilities while maintaining some of the efficiency and performance benefits of multicast service.

The field of multicast networking and related security issues is a broad technical subject. Within the space limitations allowed, we discuss some relevant technical issues and performance tradeoffs to consider when applying security and key management techniques in support of multicast networking. First, we provide a brief background of multicast technology and potential network security threats and issues. Second, we explore the application of existing and proposed security techniques for multicast networking, including key distribution, dynamic key management, and reliability issues. Throughout this paper we hope to summarize performance and security policy considerations within the context and impact of overall architectural performance.

SECURE MULTICAST GROUPS

Multicast sessions may be described in terms of their membership. In general, a session is defined as either public or private. Both types are defined by the level of session access control required to receive or transmit data within the multicast group [6]. Public sessions are typically encountered on the Internet Multicast Backbone (MBONE) and are supported by the dynamic nature of multicast communications (i.e., knowledge of the multicast address is often the only requirement for membership). Eavesdropping can quickly become a problem because of the potentially broad scope of a session. Session confidentiality can be provided through encryption. In order create a private session, access to the required session cryptographic key material should be restricted through a registration and authentication process. Only authorized users should be able to gain access to group key material and subsequently participate in the session. In this paper, we define a *secure multicast session* as a private session with encryption of data content.

SECURITY SERVICES

In order to counter the common threats to multicast communications, we can apply several of the fundamental security services, including authentication, integrity, and confidentiality as defined in [5]. A secure multicast session may use all or a combination of these services to achieve the desired security level. The amount or type of service required is dictated by the specific security policy defined for the session.

Authentication services provide assurance of a participating host identity. Authentication mechanisms can be applied to several aspects of multicast communications. Foremost, authentication is an essential part in providing access control to keying material. If the group employs cryptographic techniques such as encryption for confidentiality, then authentication measures may additionally provide a means to restricted access to the keys used to secure group communications. For an encrypted multicast session, active group membership is essentially defined by access to this keying material. Therefore, the availability and distribution of keys should be restricted to only authorized group members according to the policy of trust established for the session.

In order to identify the source of multicast traffic, *authentication* mechanisms may be applied by the traffic source. This application serves to further define group membership by positively identifying group members along with their data being sourced to the group. Protocols such as the IP Authentication Header (AH) can provide authentication for IP datagrams and may be used for host authentication [15]. Authentication is also an essential part of any key distribution

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
1998		**00-00-1998 to 00-00-1998**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Techniques and Issues in Multicast Security	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Research Laboratory,4555 Overlook Avenue, SW,Washington,DC,20375**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**20**	

protocol [16]. Because of the sensitive nature of keying material, authentication mechanisms can identify the source of the key material and provide a means to counter various masquerade and replay attacks that may be launched against a secure multicast session. Applying authentication mechanisms to transmitted multicast group data can also provide a strong level of integrity protection. Not only can these mechanisms provide a level of assurance to receivers on data origination, but they may also provide indication of data corruption.

Integrity services provide assurance that multicast traffic is not altered during transmission. Integrity is not inherent to IP datagram traffic payloads and is usually reserved for transport layer protocols (e.g., TCP). The lack or weakness of integrity services in IP can lead to spoofing attacks [17]. Strong integrity mechanisms can be applied indirectly at the network layer with security protocols such as the Encapsulating Security Payload (ESP) and AH [13, 14, 15]. In some applications where corrupted data can easily be detected, this service is not vital. However, in other applications including key management protocols, integrity services are essential means of countering spoofing attacks.

Confidentiality services are essential in creating a private multicast session. Although encryption is typically used to provide this service, a weaker form of confidentiality may be achieved by limiting data distribution of routed session IP datagrams through *time-to-live* (ttl) settings. Administrative scoping rules for multicast address spaces with a routing fabric should also be considered weak confidentiality mechanisms. Encryption can be applied at several layers of the protocol stack while maintaining the end-to-end service we desire. Multicast capable transport protocols such as RTP support encryption mechanisms within their protocol definition [8]. At the network layer, ESP provides confidentiality services for IP datagrams through encryption. Confidentiality services should also be applied to key management transactions during the exchange of key material. Key management protocols such as the Internet Security Association and Key Management Protocol (ISAKMP) [16] support confidentiality services for key exchanges. Confidentiality may also be applied to session announcements allowing them to be advertised publicly through standard methods while keeping the details of the session private.

THE APPLICATION OF SECURITY TO MULTICAST

We now present a set of multicast networking functional components and discuss relevant security issues.

Session Advertisement

Session advertisement is an important part of the overall design consideration for supporting secure multicast sessions. A generic advertisement mechanism can communicate security requirements and parameters for a secure session to its potential group members. We consider it a separate detailed policy issue whether or not the existence of a secure session is considered private information and needs confidential advertisement. In the general case, we consider it best to adapt methods already established for both non-secure and secure multicast sessions. An example uses the capabilities of the (Session Description Protocol) SDP to describe the session, and (Session Advertisement Protocol) SAP and (Session Initiation Protocol) SIP to advertise the session to group members [9, 18, 19]. This can work in a scalable manner by incorporating important session security information identified in a security association (SA) together with other non-security session essentials (e.g., start time). The SA alone typically identifies only security related parameters required to engage in a secure session [13]. The session advertisement mechanism can also serve to point potential members to a particular secure registration process if dictated by the security policy. This is a flexible way for secure multicast sessions to define unique registration processes particular to their session (such methods may often be out-of-band of the actual secure multicast data session).

Multicast Routing Protocols

In order to deliver multicast IP datagrams to group members, routers may use one of several routing protocols that define the network routing topology [2, 3, 4, 11, 22]. Some properties of these routing topologies may prove beneficial in multicast key distribution architectures and should be considered in the overall architectural picture of multicast security. For example, Ballardie presents technical key management support arguments for the Core Based Tree (CBT) multicast routing protocol [4]. Exploring the interaction of multicast security architectures and multicast routing remains an ongoing research area. In the general case, it is desirable to design multicast security mechanisms independent of any particular routing approach, as it is likely multicast routing approaches will continue to evolve.

Multicast Reliability Mechanisms

There are many multicast application classes that require a more reliable transport delivery mechanism than available through the generic and unreliable combination of UDP/IP. Key distribution is one area that benefits greatly from the introduction of efficient and reliable multicast transport methods. The overall coherence of a secure multicast session depends upon the successful distribution of keys to the secure multicast group. Military network multicast mechanisms and related application issues as discussed in [10] will likely play an important role in an overall multicast key distribution service.

Unicast design solutions of the past do not scale well to the multicast case and often present considerable efficiency concerns, exploding state maintenance, and processing burdens. Unlike unicast applications, to date there is no single reliable transport protocol like TCP that can service all classes of multicast applications [24]. For example, the reliability mechanisms used for real-time and non-real-time applications may differ because of timing constraints. In some cases, the reliability requirements of the key distribution protocol may be distinctly different from those of the application it supports. Therefore, designers should not assume a given level of reliability is always available for key distribution functions.

The security policy will dictate what reliable multicast transport mechanisms should be used to ensure that key material is delivered to all participants. In particular, the policy will dictate whether key distribution mechanisms should be sender or receiver reliable. Receiver reliable mechanisms place the responsibility of receiving the required key material on the

receiver. Sender reliable mechanisms place this burden with the distributor of the key material.

Placement of Security Mechanisms

Several technical and security policy issues must be considered prior to placing security mechanisms in the protocol stack for a secure multicast session. Varying levels of security can be achieved through placement of security mechanisms at different levels of the stack. For multicast communications, it is important to consider the impact of security mechanisms on non-security related functions at other layers of the stack, particularly reliability protocols running above UDP.

At the network layer, IP Security mechanisms can provide an important supporting role in helping to maintain secure multicast sessions. ESP and AH provide a framework for providing confidentiality, integrity, and authentication services to IP version 4 (IPv4) and IP version 6 (IPv6) protocols. Both security protocols are flexible and can support a variety of security mechanisms. They are not restricted to a specific cryptographic algorithm or other security standard. This flexibility may help resolve security implementation problems when overlapping security policies cover a multicast group [12]. For example, conflicting security policies may arise when a multicast session extends across international boundaries. In this situation, the separate policies might dictate different cryptographic algorithms with different key lengths.

In both IP security protocols, the combination of the Security Parameter Index (SPI) and its destination address uniquely identifies a particular security association [13]. In a multicast session, senders can identify a particular multicast security association using the SPI for the session and addressing its Class D address. This combination identifies the datagram as belonging to a particular multicast session but does not positively identify the originator. Because only authorized users have access to group key material, a correctly encrypted datagram is proof of membership in the group. However, in order to provide data origin authentication, a separate security association may be required for each sender to the multicast group [13]. In large groups, the additional requirement of source authentication may introduce a great deal of additional complexity to the overall system security architecture.

In some applications, network layer security may not be the best solution. Some reliable multicast protocols operating above UDP/IP may impose a level of hierarchy that may complicate a security design. For example, the Reliable Multicast Transport Protocol II (RMTP-II) builds a local recovery hierarchy at the application layer for handling receiver acknowledgments [25]. In this case, network layer security may impose some restrictions on RMTP-II unless intermediate nodes are trusted and given the group key material. Only with access to this key material can they perform their assigned duties (e.g., local caching and aggregation of control information). Therefore, it becomes a policy issue whether to extend the definition of the secure group to include others who are not the intended end-recipients of the data. Although placing security at the application layer may improve the performance of higher level protocols, it may also weaken the security of the system leaving it open to attacks which are normally protected with lower layer security (e.g., traffic analysis).

KEY MANAGEMENT ISSUES FOR MULTICAST

As introduced previously, through the use of encryption and digital signatures, we can achieve desired levels of confidentiality, integrity, and authentication for a network multicast session. Assuming the use of strong security mechanisms that cannot be easily defeated by frivolous cryptanalytic attacks, we can focus our security concerns on protecting the key material. Therefore, we focus our security concerns and the rest of our technical discussion around key management, key distribution, and access control for key material. With this in mind, a secure multicast session is defined by its Class D IP address or addresses and the required keying material.

The size, type (e.g., asymmetric vs. symmetric), and number of keys required to secure a multicast session is determined by the encryption mechanism, the employed security policies, and the keying architecture. For private multicast sessions, access to these keys must be restricted in order to maintain the security of the overall session. Therefore, during the session registration process, it is necessary to require strong authentication mechanisms to establish the identity of potential participants prior to distributing key material. When these personal attributes are bound to a signed digital certificate, the certificate's digital signature and its relationship in a certificate hierarchy [20] may verify the identity of a participant and their assigned permissions.

Depending on the network or application security policy and the amount of traffic encrypted under a particular key, it may be necessary to periodically issue a new key or "rekey" a multicast session. A rekey may also be required in the event of a suspected or detected key compromise. In this case, depending on the governing security policy, it may be necessary to exclude the compromised site from future communications. Therefore, a rekey may be targeted to specifically prohibit a compromised site from engaging in future communications without adversely affecting the rest of the group membership. Depending on the security policy in place, the definition of a compromise might include the voluntary exit of a participant from a secure session. If this occurs, the entire group may require a rekey to prevent a previous participant from rejoining the group at a later time without re-registration. In addition, the keying architecture should prevent collusion by a group of disbanded members from generating or recreating the new group key.

We note again that the proper approach and requirements for rekeying are based upon policy issues and concerns, as well as practical engineering performance tradeoffs. A policy of "flat or hierarchical" group trust may be acceptable in some scenarios greatly decreasing the complexity required for dynamic key management. Also, in some applications, the compromise of distributed keys may be an acceptable risk. Short-lived sessions with highly dynamic, but predetermined security requirements may be aptly served by a simple, flat security approach.

KEY DISTRIBUTION ARCHITECTURES

In applying a keying solution for secure multicast applications, it is desirable to maintain protocol features that preserve multicast efficiency and scale well for large one-to-many

or many-to-many data sessions. The ideal key distribution efficiency in a multicast environment can be represented in asymptotic *O-notation* as O(1). In such a scenario, a centralized server may transmit only a single keying message to the entire group to perform a group rekey. Every group member can extract the required key material from this one message. In contrast, the efficiency of using unicast techniques, without hierarchy, to distribute a group key separately to each group member is O(n). Note, in most cases, it may be more practical to perform the initial keying of participants in a unicast fashion during a registration/authentication process (this may be done out-of-band with secure e-mail, etc.). The registration function is inherently one-to-one between a single participant and the *initiator* of the session or other trusted registration authority. By coupling registration with initial key distribution, the overall number of separate transactions required can be reduced. While the initial key distribution between group members may only occur once during the lifetime of a secure session, the rekey function may occur multiple times. Therefore, it is important in some applications to focus on improving the efficiency of the rekey operation in order to improve the efficiency of the overall security solution.

Keying functions may be either centralized or distributed throughout the architecture. In a centralized architecture, keying functions are restricted to a single trusted authority. In some cases, this may be the initiator of a session or another entity assigned by the initiator to handle these vital functions. For scalability and robustness purposes, keying and registration functions may be distributed to other trusted entities. "One-to-many" type applications may benefit from a strictly centralized architecture. Alternatively, distributed architectures may prove more scalable since processing, messaging, and storage requirements are distributed across the network.

The following paragraphs provide a brief analysis of several recently proposed key distribution architectures. Each approach presents a solution to the multicast key distribution problem in a slightly different fashion. The architectures were evaluated primarily on the basis of keying efficiency and overall scalability. However, it is important to reiterate, the best solution for one particular application is often not well suited for other application or session types. For example, centralized, single source applications (e.g., multicast video servers) may benefit from a centralized keying architecture while a distributed command and control applications may benefit more from a robust, survivable distributed key distribution scheme.

Manual keying methods are often not appropriate for dynamic multicast sessions in which membership is not defined prior to the start of the session. However, in some military environments with a well-structured manual key distribution architecture already in place, this solution may be the easiest to implement.

Pairwise keying techniques similar to those presented in [7, 12, 21] typically provide linear efficiency for initial keying and rekey operations. By consolidating all rekey messages into a single multicast message, the efficiency of session rekeying can be dramatically improved. However, for *n* participants this technique increases the overall size of the rekey message to *n*. Storage requirements for pairwise techniques are minimal at participant sites but requires *n* keys to be stored with the key distributor. This method can be made more scalable if keying and registration functions are distributed to other trusted entities.

The *hierarchical trees* method presented in [12] provides linear initial keying performance and improved logarithmic rekey performance. The size[1] of any rekey message is no greater than *(k-1)d.* Key storage requirements at each participant site are *d+1* keys while the initiator must store all *key encryption keys (KEKs)* and the *group traffic encryption keys (GTEKs)*. The solution is more scalable than pairwise techniques because of the logarithmic rekey performance.

The *secure lock* method described in [23] has linear initial keying performance and an impressive constant rekey performance. The size of the rekey message is also constant providing the best rekey performance of all methods reviewed. The drawbacks of this method include the computation time for the lock and the fact that the technique is inherently centralized and may not scale well to large groups.

In order to improve overall system efficiency, the *Distributed Registration and Key Distribution (DiRK)* protocol distributes linear initial keying and rekey functions among active group members [6]. However, a question of peer trust may arise because the registration and key distribution functions are distributed in such a broad fashion. Otherwise, the solution can provide increased scalability to large networks and is appropriate in more relaxed "security compromise" environments where performance and efficiency are overriding factors.

CONCLUSIONS AND FUTURE ISSUES

IP multicast has demonstrated a capability to efficiently perform large-scale data distribution. In the basic framework, there is little or no mechanism for controlling participation within a particular multicast data session. This open framework provides useful flexibility; however, future use of secure multicast sessions in military and commercial environments is anticipated to require additional security capabilities and controls while supporting diverse policy requirements. Blindly adopting present unicast network security techniques to multicast sessions will likely sacrifice the efficient nature of multicast technology. A number of proposed promising security architecture proposals were discussed which attempt to retain message efficiency characteristics for group key distribution and key management in multicast environments.

Anticipated future multicast security requirements will be dynamic and divergent in nature requiring multiple security solutions. It is fundamentally important when addressing this problem to consider the role of related engineering and protocol performance tradeoffs. Placing unwarranted strict security policy requirements on a multicast group (e.g., lack of trust amongst keyed membership) can add significant protocol and architecture performance burdens whose tradeoffs should be carefully weighed. Individual multicast sessions and user communities will likely have different security policies based around group compromise and mutual trust. These different requirements should be taken into consideration when designing keying architectures and protocols.

[1] For a k-ary tree of depth d.

In general, several multicast security issues can be addressed through participant registration and access to multicast session keys. In many scenarios, initial participant keying is best performed out-of-band of the actual multicast data session. Subsequent key distribution can then occur within the multicast session. If compromise recovery is required within the group, several techniques described in this paper can dynamically and efficiently rekey group membership within the multicast session itself. However, as with many multicast security solutions, the best solution for one application may not be well suited for another multicast environment. Many issues including those related to security policy, multicast routing architecture, and the use of multicast reliability mechanisms may help shape a more optimal keying and security solution.

The research area of multicast security is a new and evolving field. Future solutions should consider integration with existing non-security related protocols and techniques. This includes the incorporation of reliable multicast mechanisms together with key distribution protocols. Another excellent area for future investigation in multicast security is the exploration of efficient *source authentication and integrity* for secure multicast sessions. A scalable solution should permit each group member to be identified while protecting the integrity of the user's data traffic to avoid unwarranted data injection and manipulation.

In the limited space provided, we have discussed multicast-related security issues and how multicast presents new challenges to a variety of fundamental security services. Any effective solution to a multicast security problem addresses appropriate aspects of the application of these security services without sacrificing overall engineering efficiency and scalability to large networks.

REFERENCES

[1] *Host Extensions for IP Multicast*, S. Deering, RFC 1112, 1989.

[2] *Distance Vector Multicast Routing Protocol*, S. Deering, C. Partridge, and D. Waitzman, RFC- 1075, 1 November 1988.

[3] *Multicast Extensions to OSPF*, J. Moy, RFC 1584, Proteon, Inc., March 1994.

[4] *Core Based Trees (CBT) Multicast Routing Architecture*, A. Ballardie, Internet-Draft, draft-ietf-idmr-cbt-arch-06.txt, May 1997.

[5] *Computer Communications Security: Principles, Standard Protocols and Techniques*, W. Ford, Prentice Hall, 1994.

[6] *Distributed Registration and Key Distribution (DiRK)*, R. Oppliger and A. Albanese, Proceedings of the 12th International Conference on Information Security (IFIP SEC '96), Island of Samos (Greece), May 21-24, 1996, Chapman & Hall, London, pp. 199-208.

[7] *Scalable Multicast Key Distribution*, A. Ballardie, RFC-1949, May 1996.

[8] *RTP: A Transport Protocol for Real-Time Applications*, H. Schulzrinne et al, RFC- 1889, January 1996.

[9] *SAP: Session Announcement Protocol*, M. Handley, Internet-Draft, draft-ietf-mmusic-sap-00.txt, 19 November 1996.

[10] *Reliable Multicast Data Delivery for Military Networking*, J. Macker, E. Klinker, M, Corson, *In Proceedings MILCOM 96'.*

[11] *Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification*, D. Estrin, et al, Internet-Draft, draft-ietf-idmr-pim-sm-spec-10.txt, 15 March 1997.

[12] *Key Management for Multicast: Issues and Architecture*, D. Wallner, E. Harder, and R. Agee, Internet-Draft, draft-wallner-key-arch-00.txt, 1 July 1997.

[13] *Security Architecture for the Internet Protocol*, R. Atkinson, RFC-1825, Naval Research Laboratory, August 1995.

[14] *IP Encapsulating Security Payload (ESP)*, R. Atkinson, RFC-1827, Naval Research Laboratory, August 1995.

[15] *IP Authentication Header*, R. Atkinson, RFC-1826, Naval Research Laboratory, August 1995.

[16] *Internet Security Association and Key Management Protocol (ISAKMP)*, D. Maughan, M. Schertler, M. Schneider, J. Turner, Internet-Draft, draft-ietf-ipsec-isakmp-07.txt, 21 February 1997.

[17] *Security Problems in the TCP/IP Protocol Suite*, S. Bellovin, ACM Computer Communications Review, Vol. 19, No. 2, March 1989.

[18] *SIP: Session Initiation Protocol*, Handley, Schulzrinne, Schooler, Internet-Draft, draft-ietf-mmusic-sip-03.ps, 31 July 1997.

[19] *SDP: Session Description Protocol*, M. Handley and V. Jacobson, Internet-Draft, draft-ietf-mmusic-sdp-04.ps, 2 September 1997.

[20] *Applied Cryptography, Second Edition: Protocols, Algorithms and Source Code in C*, B. Schneier, John Wiley & Sons, Inc., 1996.

[21] *Group Key Management Protocol (GKMP) Architecture*, H. Harney and C. Muckenhirn, RFC-2094, July 1997.

[22] *Protocol Independent Multicast Version 2: Dense Mode Specification*, D. Estrin, et al, Internet Draft, draft-ietf-idmr-pim-dm-05.txt, 2 April 1997.

[23] *Secure Broadcasting Using the Secure Lock*, G.H. Chiou and W.T. Chen, IEEE Transactions on Software Engineering, v. SE-15, n. 8, August 1989, pp. 929-934.

[24] *Survey on Error Recovery for IP-based Audio-Visual Multicast Applications*, G. Carle and E. W. Biersack, IEEE Network Magazine, November/December 1997, vol. II, No. 6, pp. 24-36.

[25] *The RMPT-II Protocol*, B. Whetten, et al, Internet Draft, draft-whetten-rmtp-ii-00.txt, 8 April 1998.

Techniques and Issues in Multicast Security

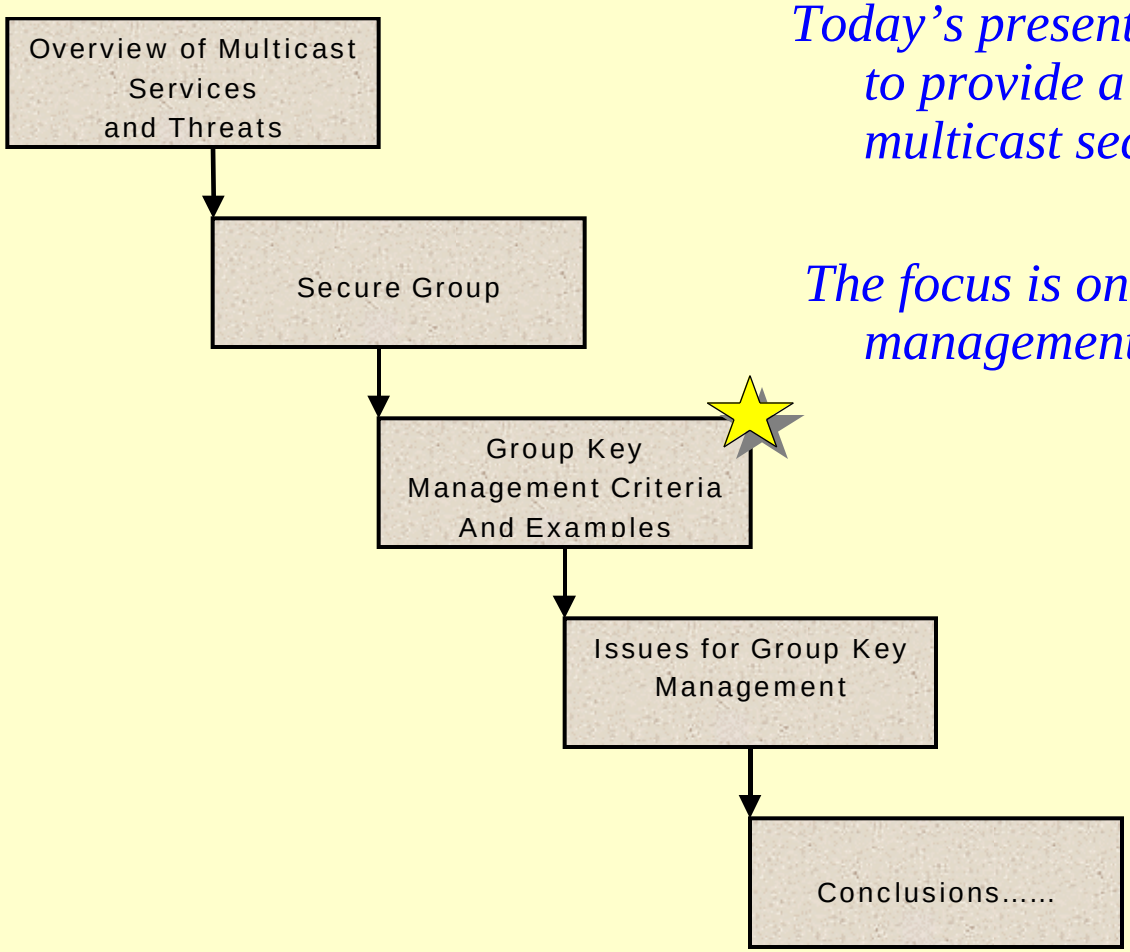
Presented for MILCOM 98

October 21, 1998

Peter S. Kruus
Naval Research Laboratory
kruus@itd.nrl.navy.mil

Joseph P. Macker
Naval Research Laboratory
macker@itd.nrl.navy.mil

Today's Presentation.....

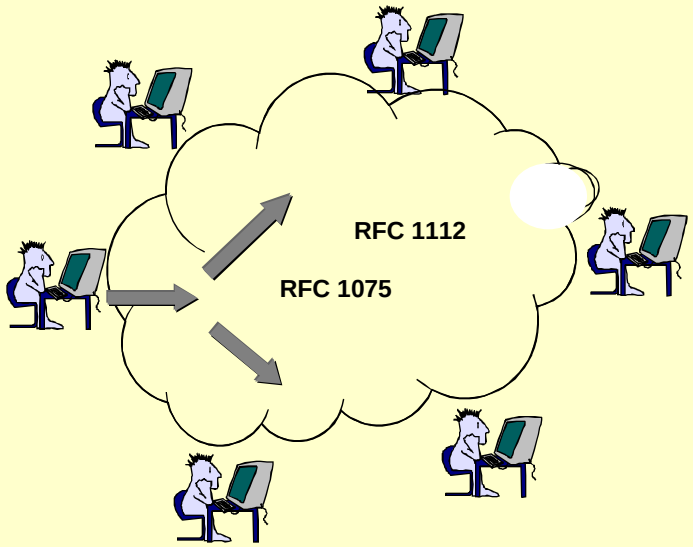


Today's presentation is intended to provide a overview of multicast security issues.

The focus is on group key management architectures.

Overview of IP Multicast Service

- IP multicast is an efficient means of distributing data to a *group* of participants.
- A sender need only transmit one copy of a datagram for the entire group.
- Multicast supports both *one-to-many* and *many-to-many* service.
- Multicast supports dynamic group communications:
  - Participants may join or leave a session at any time during its lifetime.
  - Knowledge of group's IP multicast address is required to join.



- Raw transport service is unreliable UDP/IP.
- Some RFC's which define IP multicast:
  - RFC-1112 (IP Multicast)
  - Multicast Routing: RFC-1075 (DVMRP), RFC-1584 (MOSPF), Other (e.g., CBT, PIM).

Threats to Multicast Traffic

- Multicast traffic is susceptible to the same threats as unicast traffic:
  - Eavesdropping, unauthorized creation and destruction of data, denial of service, illegitimate use of data.
- The typical security services (e.g., confidentiality, integrity, authentication) can be applied to traffic to counter these threats:
  - Security at the network layer using IPSEC mechanisms.
  - Security at the application layer for true end-to-end security.

- Because the scope of a multicast session can be large, these threats can be magnified:
  - Traffic can traverse multiple networks.
  - Large groups are more vulnerable to compromise.

- Security concerns can be abstracted into a *group key management* problem.
  - The keys used to secure the group traffic must be protected.

Secure Multicast Group

- *Participant registration and authentication mechanisms determine the type of multicast group:*
  - *Public* session often do *not* require registration or authentication. Only need IP address to join.
  - *Private* sessions require some form of registration. All participants are authenticated.
- *Secure Multicast Group* ⇨ *Private session* with encryption:
  - The secure multicast group is defined by its:
    - IP multicast address
    - Group keying material
  - The registration process defines the group by limiting access to group keying material:
    - Limit membership to paying customers
    - Limit membership to properly cleared personnel
  - Rely on strong authentication mechanisms (e.g., digital signatures) to positively identify participants.

The Secure Multicast Process

The creation and maintenance of a secure multicast session follows the following framework:

- Identify the need for a secure group.
- Define parameters for the secure session that support the group's security policy (e.g., security services, key length, crypto-algorithm).
- Determine whether assistance is required to handle registration and other keying responsibilities.
- Announce the session through posted advertisement or invitation.
- Register participants and distribute keying material.
- Perform maintenance functions including *session rekey*:
  - Rekey to replace *outdated* key material
  - Rekey to replace *compromised* key material
  - Rekey to maintain *perfect-forwards and backwards secrecy* (i.e., rekey every join and exit)

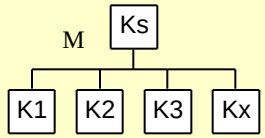
Group Key Management Criteria

Group keying schemes can be measured against the following criteria…..

- *Scalability* to support large groups (e.g., push cable application with +10,000 participants).
- *Robust* to survive link or component failures (e.g., a single key server).
- *Dynamic* rekeying to allow participants to enter and leave an active session while maintaining perfect-forwards/backwards secrecy.
- Prevention of *collusion* of disbanded participants from recreating any keying material.
- *Anonymity* in keying messages for privacy and to prevent traffic analysis.
- *Transmission efficiency* of keying messages.
- Storage *efficiency* of key material for participants and key server.
- *Computation efficiency* of key material for participants and key server.

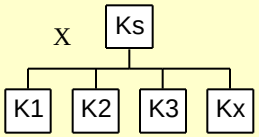
Group Key Management Architectures

Pairwise



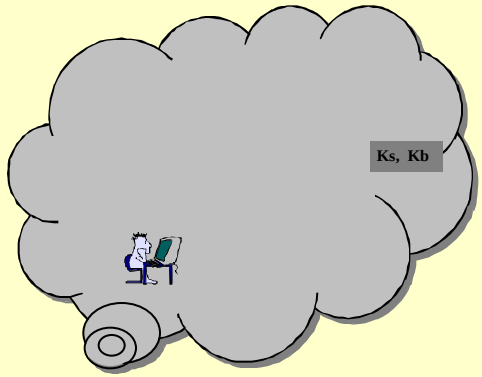
M = ({Ks}K1, {Ks}K2, {Ks}K3, ..., {Ks}Kx)

Broadcast

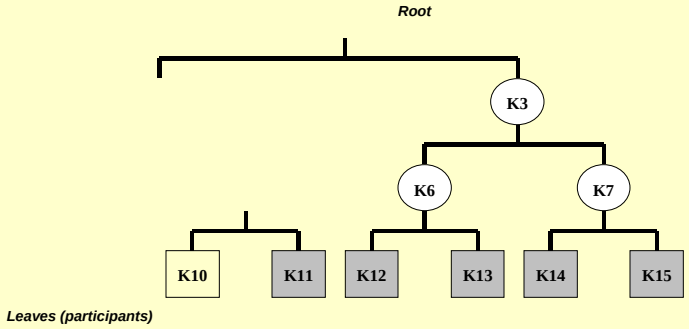


X = f ({Ks}K1, {Ks}K2, {Ks}K3, ..., {Ks}Kx)

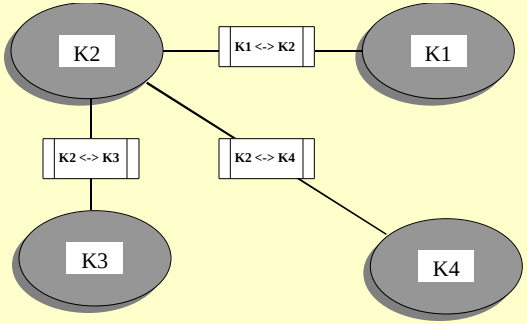
Distributed



Hierarchical



Subgroup



Other……

Comparison

Applying a strict criteria (large groups, perfect forwards/backwards secrecy):

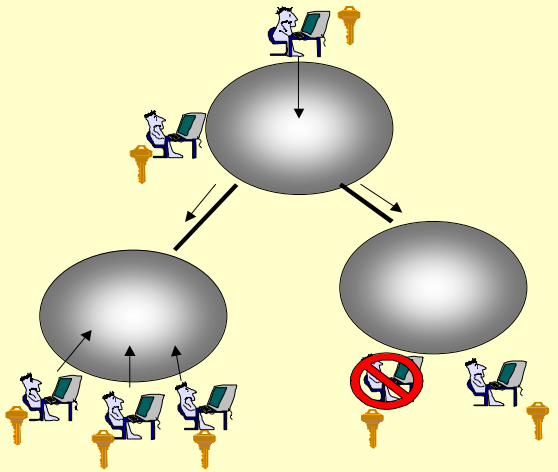
	Advantages	Disadvantages
Pairwise[1]	• Simple and straight forward approach.	• Not scalable to large groups. • Not efficient for providing perfect forwards/backwards secrecy.
Hierarchical[2]	• Scales logarithmically because of hierarchical design.	• Changes in group membership require group key to change. • Addressing required for key material.
Broadcast[3]	• Anonymity for rekey. • Common rekey package.	• Processing may approach pairwise techniques.
Distributed[4]	• Robust -> any active participant can distribute key material.	• Trust is distributed. • Membership lists or CRLs must be synchronized.
Subgroup[5]	• Membership changes only affect subgroup level.	• Architecture is not inherently robust.

Example group key architectures:

1. [GKMP]
2. [OWFT], [Wall], [Car]
3. [Lock]
4. [DiRK]
5. [Iolus]

Issues

- Multicast *security services* can suffer from scalability problems as the group size becomes large:
  - Maintaining perfect forwards/backwards secrecy becomes difficult as group *size* increases and membership *turnover rates* increases.

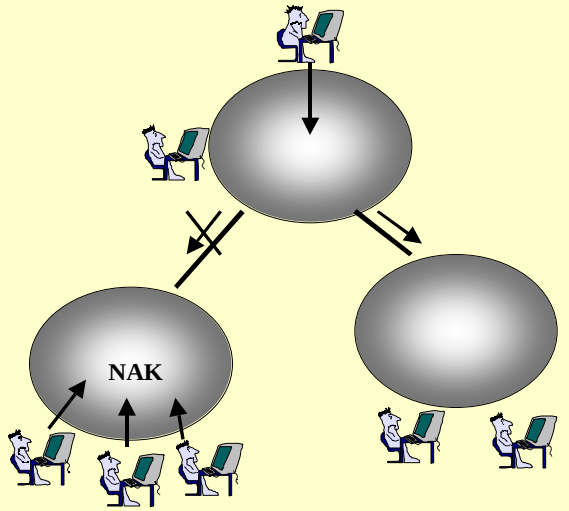


Dynamic membership creates perfect-secrecy problems.

- Centralized vs. Distributed key server:
  - *Centralized* → efficient for push applications, simpler key management, scalability problems
  - *Distributed* → robust, trust is distributed, key synchronization problems.

Issues (*continued*)

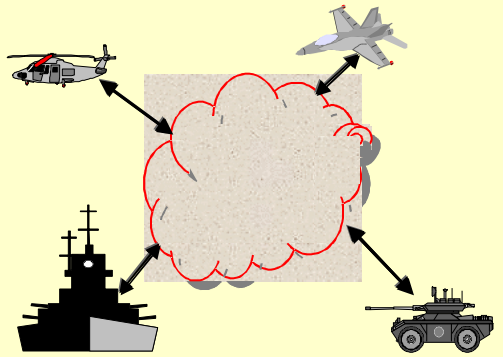
- Reliability is required for key distribution to ensure that all participants receive *rekey* material:
  - Raw IP multicast service is inherently *best effort.*
  - There are numerous reliable transport protocols that can be applied over of UDP.
  - Reliability can be either *source* or *receiver* oriented.
  - Reliable transport techniques have their own diverse performance characteristics that should be considered.
- Some reliable transport protocols can impose a hierarchy to handle requests for retransmission:
  - This hierarchy can introduce *third parties* that must be trusted by the group.



Message failures can create control message implosion problems.

Sample Keying Requirements for Tactical Military Networks

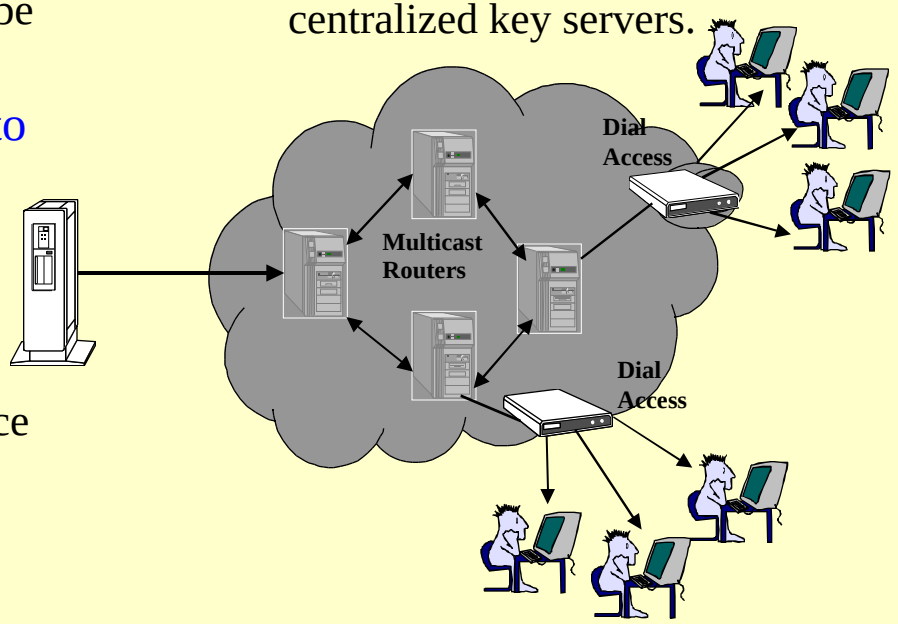
- Bandwidth constrained RF links require the *efficiency* found in multicast traffic:
  - Group key distribution should mimic multicast efficiency.
- Tactical networks must be *robust* to recover from mobile and dynamic link conditions:
  - Group key architecture should have distributed properties.
- Maintain perfect forwards and backwards secrecy:
  - Efficient rekey mechanisms.



- Participant *anonymity* required to help prevent traffic analysis:
  - Group key architecture should employ *broadcast* qualities.
- *Reliability* mechanisms are required to ensure key material is received by all participants.
- Security Services:
  - Source Authentication
  - Confidentiality, integrity

Sample Keying Requirements for Commercial Networks

- Commercial applications have potential for large groups:
  - Require a scalable solution.
- Bandwidth constrained links for dial customers:
  - Group key distribution should be efficient.
- Participant *anonymity* required to for privacy:
  - Group key architecture should employ *broadcast* qualities.
- Security Services:
  - Confidentiality, integrity, source authentication
- *Reliability* mechanisms are required to ensure key material is received by all participants:
  - The absence of multicast return channels suggests centralized key servers.



Conclusions

- Outside forces play an important role in defining an efficient key management architecture:
  - Security policy can have a defining role.
  - Other protocol layers (e.g., reliable multicast) can influence design.
- Secure multicast requires tight access control:
  - Benefits from a well established PKI.
- Any group key management solution must also consider the user application it supports:
  - Commercial push services may benefit from centralized keying schemes.
  - Tactical distributed applications may require a more robust solution.
- Reasonable solutions balance the tradeoff's for both *communications* and *security* requirements for an intended network architecture.
- In summary, there is no "one-size fits all" solution.

References

[DiRK] *Distributed Registration and Key Distribution (DiRK)*, R. Oppliger and A. Albanese, Proceedings of the 12th International Conference on Information Security (IFIP SEC '96), Island of Samos (Greece), May 21-24, 1996, Chapman & Hall, London, pp. 199-208.

[WALL] *Key Management for Multicast: Issues and Architecture*, D. Wallner, E. Harder, and R. Agee, Internet-Draft, draft-wallner-key-arch-00.txt, 1 July 1997.

[GKMP] *Group Key Management Protocol (GKMP) Architecture*, H . Harney and C. Muckenhirn, RFC-2094, July 1997.

[Lock] *Secure Broadcasting Using the Secure Lock*, G .H. Chiou and W.T. Chen, IEEE Transactions on Software Engineering, v. SE-15, n. 8, August 1989, pp. 929-934.

[Car] *Efficient Security for Large and Dynamic Multicast Groups*, G. Caronni, M. Waldvogel, D. Sun, B. Plattner, Proceedings of the Seventh Workshop on Enabling Technologies (WET ICE '98), IEEE Computer Society Press, 1998.

[Iolus] *Iolus: A Framework for Scalable Secure Multicasting*, S. Mittra, Proceedings fo the ACM SIGCOMM '97, September 14-18 1997, Cannes, France.

[OWFT] *Key Establishment in Large Dynamic Groups Using One-Way Function Trees*, D. McGrew, A. Sherman, TIS Labs at Network Associates, TIS Report #0709, 2 June 1998.